A Bit-Serial VLSI Array Processing Chip for Image Processing

ROBERT HEATON, DONALD BLEVINS, AND EDWARD DAVIS, MEMBER, IEEE

Abstract — An array processing chip has been developed integrating 128 bit-serial processing elements (PE's) on a single die. Each PE has a 16-function logic unit, a single-bit adder, a 32-b variable-length shift register, and 1 kb of local RAM. Logic in each PE provides the capability to individually mask PE's. A modified grid interconnection scheme allows each PE to communicate to each of its eight nearest neighbors. A 32-b bus is used to transfer data to and from the array in a single cycle. Instruction execution is pipelined, enabling all instructions to be executed in a single cycle. The 1-μm CMOS design contains over 1.1 million transistors on an 11.0-mm × 11.7-mm die.

I. Introduction

SCALAR machines, by their nature, process data sequentially. Their performance has typically been improved by reducing the cycle time, pipelining operation, or executing several instructions in parallel. For this reason, scalar machines will have trouble keeping up with the needs of very computation-intensive applications such as real-time image/signal processing, simulation, and modeling. Massively parallel systems, on the other hand, are well suited for these applications. Array processors achieve very high performance by exploiting the inherence parallelism of an algorithm. These machines make use of a large number of simple processing elements working in parallel to speed up a computation. In single-instruction multiple-data (SIMD) machines a single instruction is broadcast to all the processing elements (PE's) in the system simultaneously. Each PE then performs the global operation on its own data. Massively parallel machines typically support from 1K to 16K PE's. In most cases, a PE utilizes a single-bit ALU and is connected to its neighbors through a bit-serial grid, hypercube, or other form of communication network. A number of massively parallel machines have been built, including the Massively Parallel Processor (MPP) built for NASA by Goodyear Aerospace, the Distributed Array Processor (DAP) built by AMT, and Thinking Machines' Connection Machine [1], [2]. To date, such systems have primarily been used by research organizations and the military. With reductions in hardware cost and the development of better parallel programming languages and new algorithms, parallel machines will be able to provide competitive, cost-effective performance.

Manuscript received August 24, 1989; revised December 4, 1989. This work was supported in part by a grant from NASA's Goddard Space Flight Center.

R. Heaton is with the Microelectronics Center of North Carolina, Research Triangle Park, NC 27709.

D. Blevins was with the Microelectronics Center of North Carolina, Research Triangle Park, NC. He is now with Precision Products Corporation, Lexington, KY 40508.

E. Davis is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695.

IEEE Log Number 8934063.

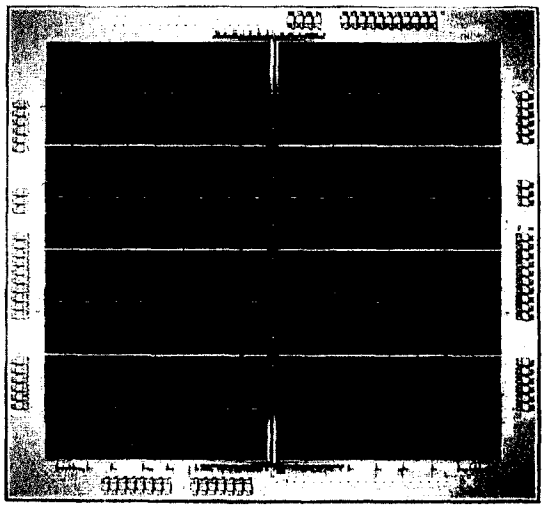

Fig. 1. Chip photomicrograph.

II. BLITZEN Chip Architecture

Each chip contains 128 PE's arranged as an 8 × 16 grid [3], [4]. It is implemented in 1-μm double-level-metal CMOS and contains 1.1 million devices (see Figs. 1 and 2 and Table I). Internally, a three-stage pipeline enables BLITZEN to execute an instruction every cycle, as shown in Fig. 3. During the first cycle, an instruction from the external control unit is latched and decoded. The instruction contains a 23-b opcode, a 10-b memory address, and a 4-b column select address. The 23-b opcode is broken into a number of independent subfields. Each subfield controls a particular function of a PE, enabling the programmer to perform several suboperations simultaneously (see Fig. 4). Internally, the 23-b opcode is decoded into a fully horizontal 59-b microinstruction. During the second stage of the

0018-9200/90/0400-0364$01.00

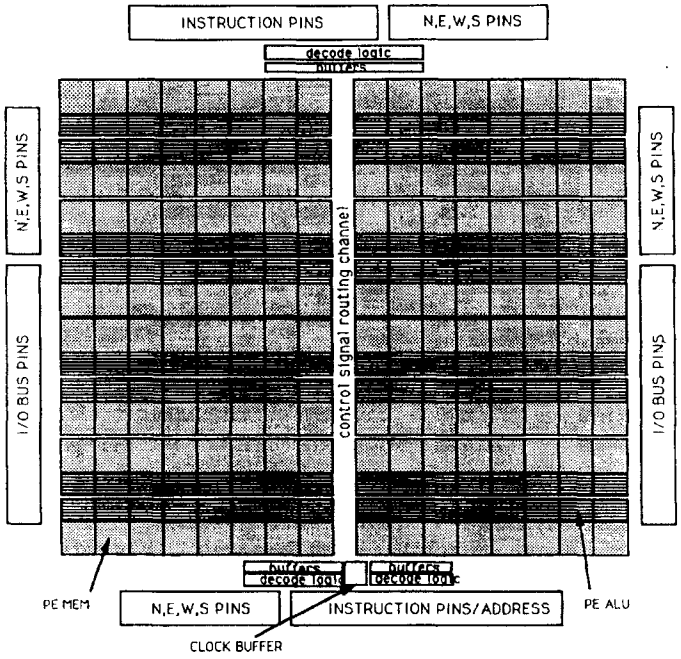

Fig. 2. Chip block diagram.

TABLE I

PROCESSING DESIGN RULES AND ELECTRICAL SPECS

transistors	1,109,340	die size	11.0 x 11.7mm
Power Supply	3.3 ± .3 V	operating frequency	20MHz.
Package	176 pin PGA	power dissipation	0.7 Watts at 0°C
Gate Length	1.0 μm	Poly Pitch	2.6 μm
Metal1 Pitch	2.6 μm	Metal2 Pitch	4.0 μm
Tox	22.5 nm		

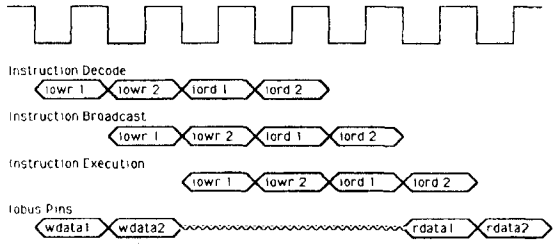

Fig. 3. Instruction pipeline for IOBUS transfers.

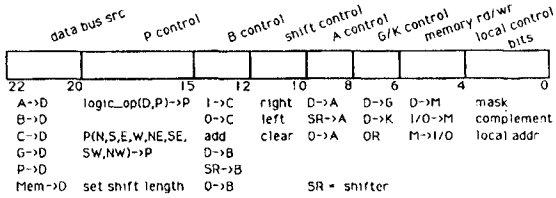

Fig. 4. Instruction format.

pipeline, the microinstruction is buffered and broadcast to all of the PE's. In the final stage, the instruction is executed.

A grid network is provided enabling each PE to transfer data, bit serially, to its nearest neighbors. This operation is done in parallel by all PE's, allowing data to be quickly shifted across the array. Most parallel machines provide connections to the neighbors in the four primary directions (north, south, east, and west). BLITZEN PE's are connected together in an "X" configuration (see Fig. 5), enabling each PE to communicate directly with its eight nearest neighbors (four primary directions as well as the four diagonals). For example, data can be routed to the north by transmitting on the northeast grid connection and receiving on the southeast. Similarly, a transfer in the southwest direction can be achieved by transmitting from the southwest port and receiving on the northeast port. The X grid routing network, as it is known, is extended off chip, so that a system containing an array of chips can be uniformly interconnected. All routing instructions are executed in a single clock cycle. At the boundaries of the array, off-chip logic can be used to reconfigure the two-dimensional routing network to form a long linear array of PE's or more exotic configurations such as a cylinder or a torus.

Data transfers to and from the array are often a bottleneck in parallel systems. The BLITZEN I/O scheme was designed to interface to the new generation of video DRAM's which are capable of very high block transfer rates. The 32-b IOBUS is internally divided into eight 4-b buses, one for each of the eight rows of PE's (see Fig. 5). Each 4-b bus is shared by the 16 PE's in a row. IOBUS transfers take place directly between external video DRAM

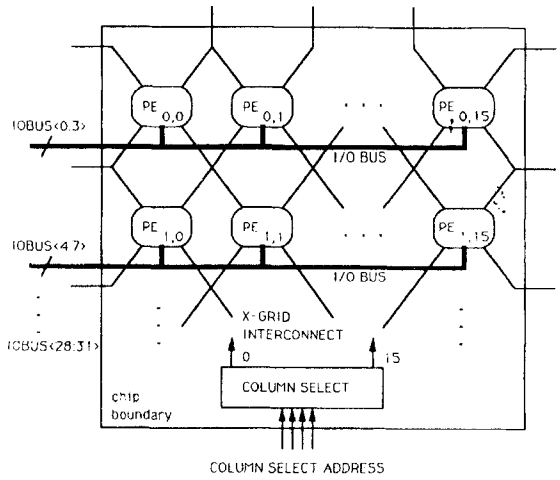

Fig. 5. PE interconnections.

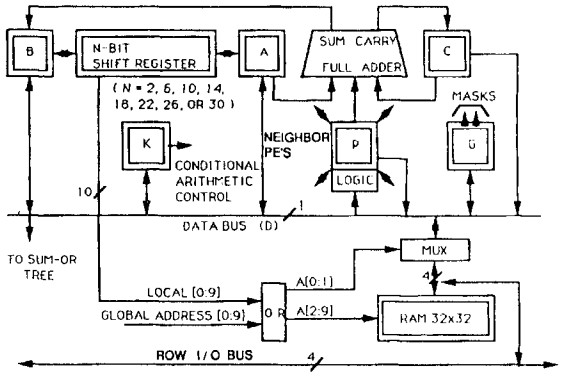

Fig. 6. PE block diagram.

and a PE's local memory. During an IOBUS transfer, four column select bits, issued along with the instruction, are used to select one of the 16 columns of PE's. The eight most-significant bits of the instruction's address field identify the 4-b nibble in each PE's local RAM, which will be involved in the transfer. In this way, 32 b of data can be read/written to a column of PE's in a single cycle (see Fig. 3).

The subfields of the instruction opcode are arranged such that I/O transfers and routing operations can be performed in parallel with other PE operations. This enables the user to hide the I/O and data routing time behind the execution of other array computations.

III. BLITZEN PE Architecture

Each PE in BLITZEN is a bit-serial processor. Fig. 6 shows the functional elements of a single BLITZEN PE. There are six single-bit registers: A, B, C, G, K, and P. Each of the six registers is attached to the local data bus D. A 16-function logic unit allows the user to perform logical operations between the current contents of the P register and the value on the data bus, with the results being stored in P. Routing operations are also done via the P register. An arithmetic unit enables the user to perform single-bit full or half addition, with the sum stored in B and the carry saved in C. A bidirectional shift register is used as an accumulator to hold arithmetic operands. Combined with the A and B registers the shifter can be configured to hold 4-, 8-, 12-, 16-, 20-, 24-, 28-, or 32-b values. An N-bit add can then be accomplished using the following instruction sequence:

N bit add of two operands op1 and op2 residing in local memory with the result left in the shift register

load op1 into shifter

for($i = 0$; $i < N$; $i++$)

{memory location (op1 + i) → B, shift right}

first bit of op2 is loaded into P

{memory location (op2) → P}

in parallel:

1. *the contents of A and P are added*
2. *the shifter is shifted right moving the next bit into A and storing the first bit of the result in B*
3. *the next bit of op2 is loaded into P*

for($i = 1$; $i < N$; $i++$)

{memory location (op2 + i) → P, $A + P \rightarrow B$, shift right}

result left in the shifter

The operations in { }s can be implemented in a single instruction. As a result, the sequence above would require N cycles to load the *op1* into the shift register and N cycles to perform the addition.

SIMD machines broadcast a single common instruction to all PE's in the system. In many cases it is useful to locally modify the instruction seen by the PE's. For example, if one wanted to perform the operation

$$\text{if}(A[i][j] > 6)\, B[i][j] = B[i][j] + 1$$

it would be necessary to determine for which PE's A was greater than 6 and then selectively increment B for those PE's. The K and G registers in each PE are provided for this purpose. When a "maskable" instruction is issued, all PE's whose G bits are set to zero will be masked and will not perform the broadcast operation. All other PE's, with their G registers set, will execute the instruction normally (see Fig. 7). Using this approach, PE's can be selectively enabled and disabled.

The K register is used to perform another type of conditional operation. When a "complementable" instruction is broadcast, all PE's whose K bits are cleared will do the complement of the logic or arithmetic function specified. Those PE's which have their K bits set will perform the normal operation while the remaining PE's will do its complement. Using the K register, designated PE's can be performing addition while others are doing subtraction.

This feature can be used, for example, to improve the execution time of nonrestoring division by a factor of 2

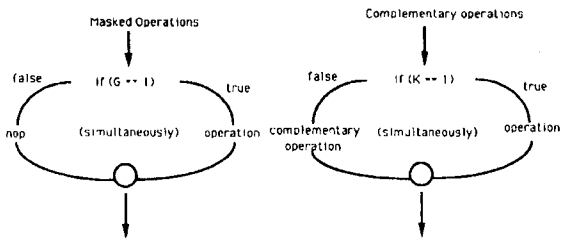

Fig. 7. Local PE control: masked and complementable operations.

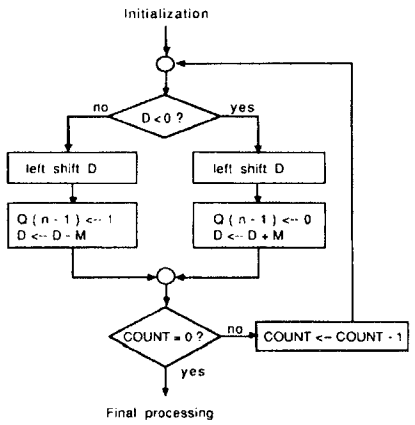

Fig. 8. Iterative loop for nonrestoring division.

(see Fig. 8). The nonrestoring division algorithm consists of an iterative loop which depends on the results of the previous step. If the previous step produced a negative result, then during the current iteration a quotient bit of zero will be generated and the divisor is added to the existing partial remainder. If the previous step produced a positive partial remainder, a quotient bit of one is generated and the divisor is subtracted from the partial remainder. Without the K register, the two parallel paths in the division loop must be executed sequentially. First, a subtraction iteration is performed in the PE's that have a positive partial remainder while the other PE's are masked, followed by an addition in those processers that have a negative partial remainder. The two branches of the division loop happen to be complements of each other. By storing the sign of the partial remainder in the K register, PE's with positive partial remainders can follow the left branch while the remaining PE's can be concurrently following the right path. When not being used during conditional instructions, K and G serve as regular data registers.

Each PE also has 1024 b of local static RAM. The address is provided by a 10-b field in the user instruction and is broadcast to all the PE's along with the opcode. A READ operation places the contents of the specified memory location on the internal data bus. A WRITE saves the current value of the data bus in the memory. A READ or WRITE operation occurs in a single cycle. The local RAM can be thought of as a register set and enables the user to cache frequently used operands for quick access. BLITZEN is different than most other massively parallel SIMD machines in that the global address, broadcast to all the PE's, can be locally ORed with the ten most significant bits of the shift register. Using local addressing, each PE can further customize its operation. By broadcasting the base address of a table, each PE can be using different data values from the table in a common calculation.

An OR tree is connected to all PE's on the chip, enabling the values presented on each of the 128 PE data buses to be ORed together. This feature enables the user to quickly test for a "true" bit in any of the PE's of the array. The OR tree is useful in associative operations and in performing data searches. OR tree operations are pipelined. The single OR pin output is open drain, enabling several BLITZEN chips to be directly wire ORed together.

IV. Conclusion

This paper has reported on the architecture and design of a 1.1-million-transistor VLSI array processing chip. The chip forms the core of a highly integrated, massively parallel machine. A 16K PE system (128 BLITZEN chips), operating at 20 MHz, is capable of performing IEEE single-precision multiplication at a rate of 450 MFLOPS. Over one billion operations per second can be achieved for 32-b fixed-point arithmetic. Since processing is bit serial, proportional speed improvements can be obtained with shorter word lengths. The I/O scheme is capable of transferring data at a rate of 10 gigabytes per second.

References

[1] W. D. Hillis, *The Connection Machine*. Cambridge, MA: M.I.T. Press, 1985.

[2] K. E. Batcher and J. L. Potter, Eds., "Array unit," in *The Massively Parallel Processor*. Cambridge, MA: M.I.T. Press, 1985.

[3] D. W. Blevins, E. W. Davis, and J. H. Reif, "Processing element and custom chip architecture for the BLITZEN massively parallel processor," Microelectronics Center of North Carolina, Research Triangle Park, Tech. Rep. TR87-22, revised, June 10, 1988.

[4] D. W. Blevins, E. W. Davis, R. A. Heaton, and J. H. Reif, "Blitzen: A highly integrated massively parallel machine," in *Proc. 2nd Symp. Frontiers of Massively Parallel Computation*, Oct. 12, 1988.

Robert Heaton graduated from Brown University, Providence, RI, in 1980 and received the M.S. degree in electrical engineering from Carnegie-Mellon University, Pittsburgh, PA, in 1982.

In 1980 he became a Member of the Technical Staff at AT&T Bell Laboratories. While there he worked on the design of AT&T's WE 32000 CPU family and the CRISP microprocessor. He has been a Member of the Technical Staff with the Microelectronics Center of North Carolina, Research Triangle Park, since June 1987. His interests include the design of high-performance VLSI circuits and the development of CAD tools.

Donald Blevins received the M.S. degree in electrical engineering from Rutgers University, New Brunswick, NJ, in 1986.

In 1984 he become a Member of the Technical Staff at AT&T Bell Laboratories. While there he worked on the design of the WE 32301 Memory Management Unit and a graphics processor. In 1986 he became a Member of the Technical Staff at the Microelectronics Center of North Carolina. He is now with Precision Products Corporation in Lexington, KY.

Edward Davis (S'62–M'72) received the B.S. and M.S. degrees in electrical engineering from the University of Akron, Akron, OH, in 1964 and 1967, respectively, and the Ph.D. degree in computer science from the University of Illinois, Urbana, in 1972.

He worked for Goodyear Aerospace Corporation (now LORAL Defense Systems, Akron) on the STARAN parallel processor. He is currently an Associate Professor of Computer Science at North Carolina State University, Raleigh, with interests in computer architecture and parallel processing.